AD-A082 847 PRINCETON UNIV NJ DEPT OF STATISTICS F/G 9/4

AN INDUCTION THEOREM FOR DISCOVERING SYNTACTIC TRANSLATIONS.(U)

JAN 80 P R THRIFT N00014-79-C-0322

UNCLASSIFIED TR-160-SER-2 NL

1 of 1

AD A
082847

END

DATE
FILMED

5-80

DTIC

AN INDUCTION THEOREM FOR DISCOVERING SYNTACTIC TRANSLATIONS.

by

Philip R. Thrift
Princeton University

A

Technical Report No. 160, Series 2
Department of Statistics
Princeton University
January 1980

Research supported in part by a contract with the Office of Naval Research, No. N00014-79-C-0322, awarded to the Department of Statistics, Princeton University, Princeton, New Jersey.

ABSTRACT

Given an input-output sequence of syntactic translations of sentences generated by a deterministic finite state grammar G into Σ^*, a method is given for discovering the function which maps productions of G into Σ^* that gives rise to the observed translation.

---

1. INTRODUCTION

Let $G = (V_N, V_T, P, S)$ be a right linear grammar [2]. Thus all productions in P are of the form

$$A \to aB \qquad \text{or} \qquad A \to a$$

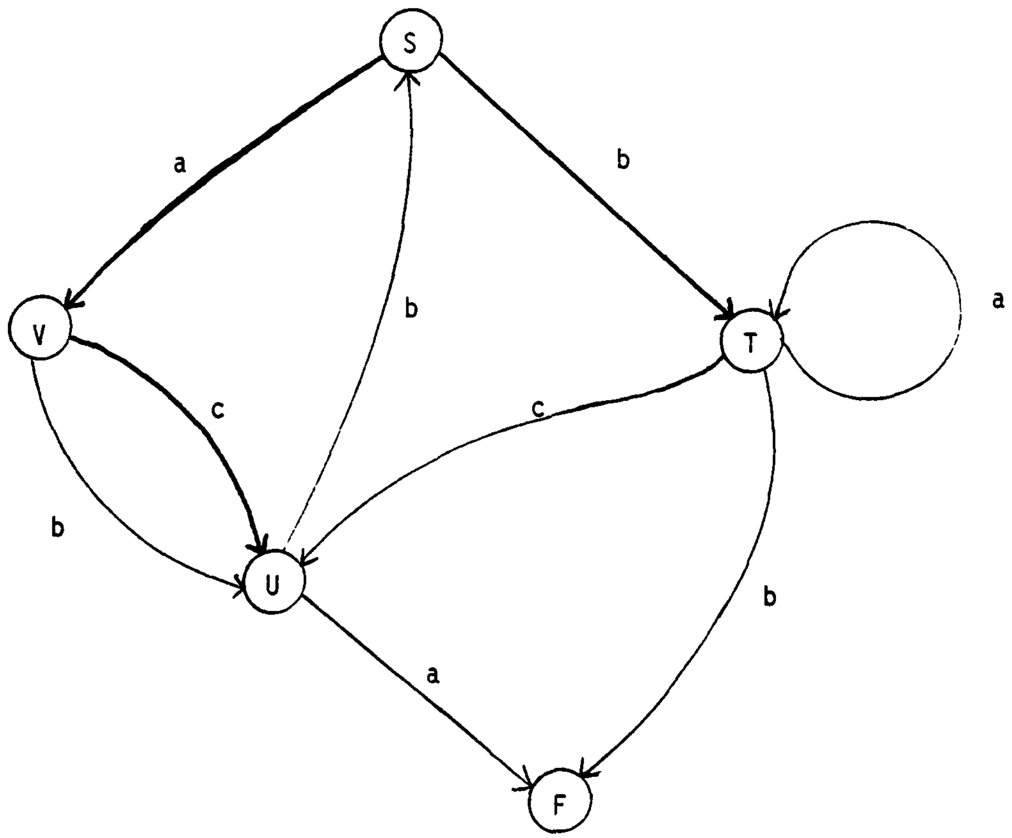
where A and B are syntactic variables in V_N, and a is a terminal (or word) in V_T. We shall assume that G is *deterministic*, by which we mean that for every pair $(A, a) \in V_N \times V_T$ there is at most one production in P of the above form. We denote the set of sentences generated by G by L(G).

With G we shall associate what we shall call the *wiring diagram* $\mathcal{G}$ of G.

A

<u>Definition</u>. Let G be a right linear grammar. Then the wiring diagram $\mathcal{G}$ of G is a directed pseudograph [3] with labelled arcs. The node set $N(\mathcal{G})$ is $V_N \cup \{F\}$, where F is a symbol not in $V_N \cup V_T$. The arc set $A(\mathcal{G})$ is determined by the productions of G: if $A \to aB$ is an element of P then $A \xrightarrow{a} B$ is a labelled arc of $\mathcal{G}$; if $A \to a$ is an element of P then $A \xrightarrow{a} F$ is a labelled arc of $\mathcal{G}$.

For example, if $G = \left\{\{S, T, U, V\}, \{a, b, c\}, P, S\right\}$ where $P = \{S \to aV|bT,\ T \to aT|cU|b,\ U \to bS|a,\ V \to cU|bU\}$, then $\mathcal{G}$ is shown in Figure 1.



<u>Figure 1.</u>

There is obviously a natural correspondence between the elements of $L(G)$ and the set of walks from S to F in G; i.e., $L(G) = \{x_1 \ldots x_n | S \xrightarrow{x_1} X_1, X_1 \xrightarrow{x_2} X_2, \ldots, X_{n-1} \xrightarrow{x_n} F$ are labelled arcs of G, for some $X_1, \ldots, X_{n-1} \in V_N\}$. We shall assume throughout this paper that for each $A \in V_N$ in G there is a path from S to F that passes through A.

<u>Definition</u>. Given a deterministic right linear grammar G and a finite abstract set of symbols $\Phi = \{\phi_1, \ldots, \phi_s\}$, a <u>syntactic translation</u> is a map f from $A(G)$ to Φ^*.

If $A \xrightarrow{a} B$ is a labelled arc of G and if the image of this arc under f is ϕ where $\phi \in \Phi^*$, then graphically we write

$$A \xrightarrow{a|\phi} B$$

(Φ^* is the set of finite length sequences from Φ, including Λ, the empty string).

This definition is basically equivalent to the definition of a <u>generalized sequential machine</u> (gsm) [1], where f is called an <u>output function</u>.

By extending the definition of f in the natural way we have

$$f^{ex}: L(G) \to \Phi^* ;$$

i.e., if we have under f

$$S \xrightarrow{a_1|\phi^{(1)}} A_1, \ldots, A_{n-1} \xrightarrow{a_n|\phi^{(n)}} F$$

with $\phi^{(1)}, \ldots, \phi^{(n)} \in \Phi^*$, then the sentence

$$a_1 a_2 \ldots a_n \xrightarrow{f^{ex}} \phi^{(1)} \phi^{(2)} \ldots \phi^{(n)} .$$

In the syntactic translation as shown in Figure 2,

$$ba^2b \rightarrow \phi_5\phi_4\phi_5\phi_5\phi_4\phi_1$$

$$acbaba \rightarrow \phi_3\phi_1\phi_3\phi_1\phi_2\phi_3\phi_1\phi_3\phi_1\phi_1\phi_2 \text{ ,}$$

etc.

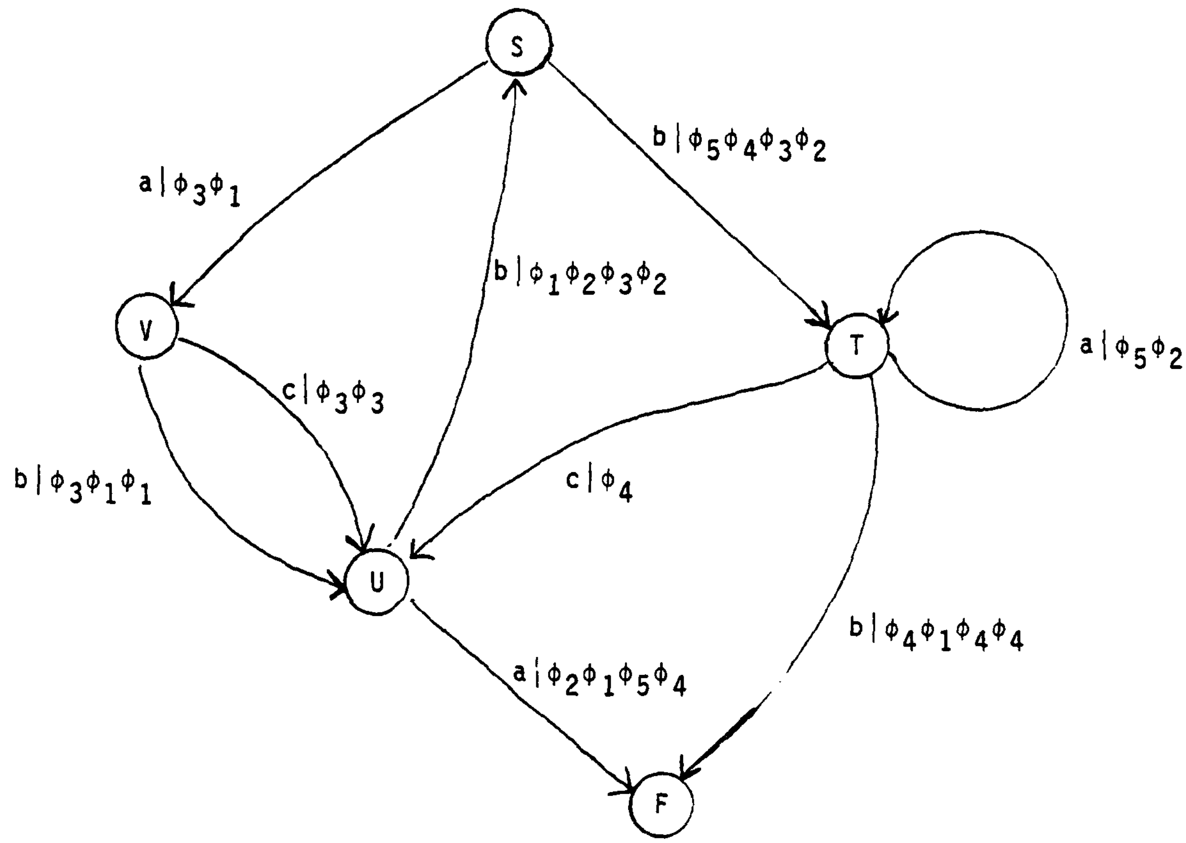


Figure 2.

Let $A(G, \Phi^*)$ be the set of syntactic translations of G, and let $A^{ex}(L(G), \Phi^*)$ be the extension of A to $(\Phi^*)^{L(G)}$. We shall refer to elements of $A^{ex}(L(G), \Phi^*)$ as <u>syntactic maps</u>.

2. TREE COMPOSITIONS

Definition. Let Σ be a finite alphabet, and $x \in \Sigma^*$. A k-composition of x is defined to be an ordered k-tuple $c \in (\Sigma^*)^k$, $c = (c_1,\ldots,c_k)$ having the property that $c_1c_2 \ldots c_k = x$. The set of k-compositions of x is denoted $C_k(x)$.

For example, if $\Sigma = \{a,b,c\}$, then $C_3(ab^2c)$ is the set $\{(\Lambda,\Lambda,ab^2c), (\Lambda,a,b^2,c), (\Lambda,ab,bc) \ldots\}$ where Λ denotes the empty word. In general, $|C_k(x)| = \binom{n+k-1}{k-1} = \binom{n+k-1}{n}$ if $|x| = n$.

The notion of composition is extended to trees.

Definition. Let Σ be a finite alphabet, T a rooted directed tree $T = (N(T), A(T))$. Thus T is a directed tree with a distinguished node $R \in N(T)$, and for each node $N \in N(T)$ there is a unique directed path from R to N. The leaves of T, denoted $L(T) \subset N(T) - R$ are the nodes of T with degree 1. Assume the elements of $L(T)$ are ordered $L_1,\ldots,L_\ell$ where $\ell = |L(T)|$. For a given element $x = (x_1,\ldots,x_\ell) \in (\Sigma^*)^\ell$ a T-composition of x is defined by a function

$$A(T) \xrightarrow{t^c} \Sigma^*$$

having the property that for each leaf L_j of T, and unique path $a_1,\ldots,a_k \in A(T)$ from R to L_j,

$$t^c(a_1)t^c(a_2) \ldots t^c(a_k) = x.$$

Thus a tree composition reduces to a k-composition when the tree is a rooted path consisting of k connected arcs. An example of a tree composition of (ab,ab,b,ba) is shown in Figure 3, for the complete binary tree with 7 nodes. Given T, along with an ordering for the leaves, and $x \in (\Sigma^*)^{L(T)}$ we denote the set of all tree compositions of x by $TC(T,x)$.

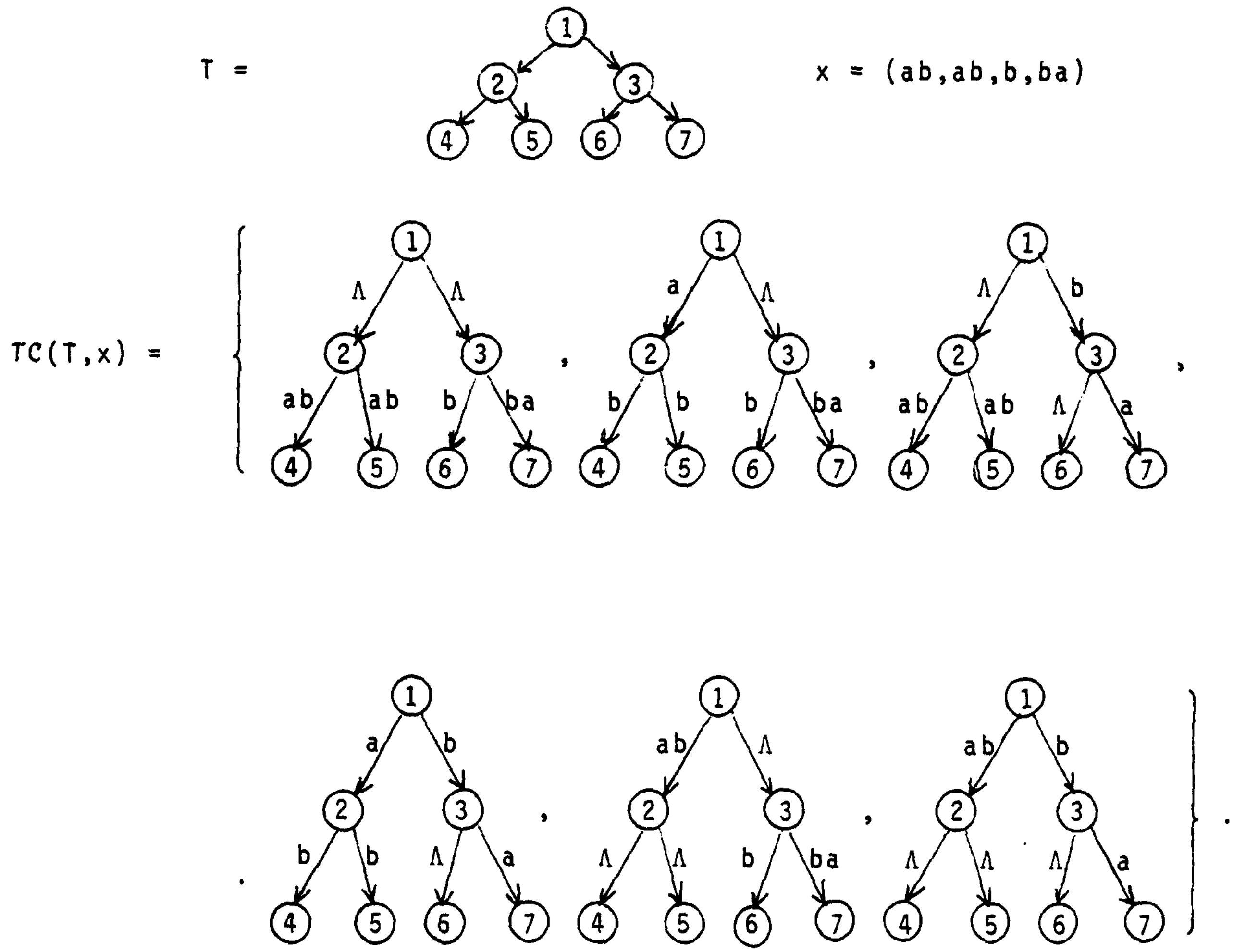


Figure 3.

An element of $TC(T,x)$ can be represented as a non-negative integer lattice point in a natural way:

If $a_1, \ldots, a_{|A(T)|}$ is some ordering of the arcs, then

$$t^c(a) \longrightarrow |t^c(a)| \qquad a \in A(T)$$

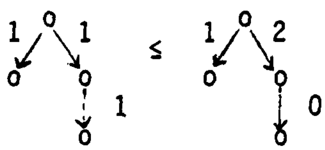
specifies a lattice point in $L = \mathbb{N}^{|A(T)|}$, $\mathbb{N}$ = non-negative integers.

We denote by $S[TC(T,x)]$ the set of lattice points defined above. A partial order $\leq_T$ is defined in L : for $s,t \in L$

$$s \leq_T t \quad \text{iff} \quad t \text{ is obtained from } s$$

by moving objects up the tree.

For example,



$$, \quad o \xrightarrow{1} o \xrightarrow{1} o \;\leq\; o \xrightarrow{2} o \xrightarrow{0} o \text{ , etc.}$$

We define, for $S \subset L$, max S = the elements of S having the property that for no $t \in S$: $s \leq t$, $t \neq s$.

3. THE INDUCTION PROBLEM

It is possible for two distinct syntactic translations to be extended to the same syntactic map. Thus we define an equivalence relation, ~ , on $S(G,\Phi^*)$ by defining $f_1 \sim f_2$ iff f_1 and f_2 are extended to the same element of $S^{ex}(L(G),\Phi^*)$.

The induction problem for syntactic translations is this: an observer O, who we assume knows the internal structure of the wiring diagram G except for the syntactic translation, can observe sentences from $L(G)$ along with their image in Φ^* under the unknown syntactic translation. Thus he can observe the syntactic map for a few sentences in $L(G)$. O wishes to discover an element $f \in S(G,\Phi^*)$ (up to equivalence) such that f^{ex} holds. We assume O can pick the sentences he wishes to observe. The theorem that follows shows, essentially, that O can pick a finite number of sentences from L(G) from which syntactic translation discovery is possible.

THEOREM: The syntactic translation (up to equivalence) can be discovered by observing a finite number of sentences W.

Remark: What the theorem says is that on observing a finite set W (to be constructed below), $\mathcal{O}$ is presented with a finite number of word equations:

$$(E)\qquad \begin{aligned} &a_{11}a_{12}\cdots a_{1i_1} = \phi(1)\\ &\vdots\\ &a_{k1}a_{k2}\cdots a_{ki_k} = \phi(k)\end{aligned}$$

where $|W| = k$, $a_{mn} \varepsilon A(G)$ (the arc set of G) and $\phi_{(j)}$ the observed image in Φ^* corresponding to the sentence determined by the walk $a_{j1} \ldots a_{ji_j}$ in G. A solution of E (that is, an assignment of values in Φ^* to the arcs $A(G)$ so that E is satisfied) will solve the induction problem.

Proof: The proof follows the construction of the implicit functions in [4].

We construct a spanning tree T in G, rooted at S and connecting all nodes in V_N. F is not connected to the spanning tree. For the example of Figure 1, a spanning tree T is indicated by darkened lines.

Label the arc set $A(G)$ in such a way that $A(T)$, the set of arcs in the spanning tree are $a_1,\ldots,a_t$.

From Φ and $A(T) = \{a_1,\ldots,a_t\}$ we create a new set of symbols. In general let X be a finite alphabet $\{x_1,\ldots,x_n\}$. Then define X^o *to be the group freely generated by the symbols of* X, with Λ the identity element. Form $(\Phi \cup A(T))^o$.

Begin at F and consider all arcs a entering F. Call this set $A(F)$, $A(F) \neq \phi$. Take an element a in $A(F)$. In what follows if a is the arc $A \xrightarrow{x} F$ then $\alpha(a) = A$, $\omega(a) = F$. Thus $\alpha(a) \in V_N$ and thus there is some walk $w = a_{i_1},\ldots,a_{i_j}, a$ from S to F with $a_{i_1},\ldots,a_{i_j} \in A(T)$. The sentence determined by the walk w, call it s, is mapped to $\phi(s)$, which $\mathcal{O}$ observes and writes

$$a = a_{i_j}^{-1} \ldots a_{i_1}^{-1}\phi \in (\Phi \cup A(T)^o) .$$

This is done for each element of $A(F)$.

$\mathcal{O}$ now considers the arcs of $A(G) - (A(T) \cup A(F))$. Let $A^{(j)} =$ the set of arcs a of G not in $A(T)$ such that the number of arcs in the shortest path (a walk with no repeated nodes) from $\omega(a)$ to F is j (i.e., $A^{(0)} = A(F)$). Suppose $\mathcal{O}$ has computed the equations for the arcs in $A^{(0)},\ldots,A^{(j-1)}$. Let $a \in A^{(j)}$ and let $a,b_1,\ldots,b_j$ be a shortest path from $\omega(a)$ to F. Now $\alpha(a) \in V_N$ hence

$$a_{i_1} \ldots a_{i_j}\, \underline{a}\, b_1 \ldots b_j,$$

a walk from S to F, $a_{i_1},\ldots,a_{i_j} \in A(T)$. If this corresponds to sentence s then $\mathcal{O}$ observes $\phi(s)$, so that $a = a_{i_j}^{-1} \ldots a_{i_1}^{-1}\, \phi\, b_j^{-1} \ldots b_1^{-1} \in (\Phi \cup A(T))^o$ by using the equations for $b_1,\ldots,b_j$ from previous computations. This process terminates with a list of equations

$$(I)\begin{cases} a_{k+1} = g_1 \\ \vdots \\ a_q \quad = g_{q-k} \end{cases}$$

where $g_1, \ldots, g_{q-k}$ are elements of $(\Phi \cup A(T))^0$.

For example, from Figure 3 if we define the arcs

$$\left.\begin{array}{ll} a_1 & S \xrightarrow{a} V \\ a_2 & S \xrightarrow{b} T \\ a_3 & V \xrightarrow{c} U \end{array}\right\} \text{ spanning tree} \;.$$

$$\begin{array}{ll} a_4 & U \xrightarrow{a} F \\ a_5 & T \xrightarrow{b} F \\ a_6 & T \xrightarrow{a} T \\ a_7 & T \xrightarrow{c} U \\ a_8 & U \xrightarrow{b} S \\ a_9 & V \xrightarrow{b} U \end{array}$$

Then

$$\begin{aligned}
a_1 a_3 \underline{a_4} &= \phi_3 \phi_1 \phi_3^2 \phi_2 \phi_1 \phi_5 \phi_4 \\
a_2 \underline{a_5} &= \phi_5 \phi_4 \phi_3 \phi_2 \phi_4 \phi_1 \phi_4^2 \\
a_2 \underline{a_6} a_5 &= \phi_5 \phi_4 \phi_3 \phi_2 \phi_5 \phi_2 \phi_4 \phi_1 \phi_4^2 \\
a_2 \underline{a_7} a_4 &= \phi_5 \phi_4 \phi_3 \phi_2 \phi_4 \phi_2 \phi_1 \phi_5 \phi_4 \\
a_1 \underline{a_9} a_4 &= \phi_3 \phi_1 \phi_3 \phi_1^2 \phi_2 \phi_1 \phi_5 \phi_4 \\
a_1 a_3 \underline{a_8} a_2 a_5 &= \phi_3 \phi_1 \phi_3^2 \phi_1 \phi_2 \phi_3 \phi_2 \phi_5 \phi_4 \phi_3 \phi_2 \phi_4 \phi_1 \phi_4^2 \;.
\end{aligned}$$

These equations can be solved in the group $(\Phi \cup A(T))^0$ by the method indicated.

It follows from [4] that, given (I), the syntactic map is the same for all assigments of $a_1, \ldots, a_k$ to elements of Φ^0, and hence Φ^*. What this means is that, given the finite equations (I), an assignment of values in Φ^* to the arcs of the spanning tree $a_1,\ldots,a_k$ so that $a_{k+1},\ldots,a_q$ as defined by (I) are in Φ^* will solve the induction problem. [].

A sequence $a_1,\ldots,a_k \in \Phi^*$ such that $a_{k+1},\ldots,a_q$ are in Φ^* is called a <u>feasible point</u>.

4. THE INDUCTION SOLUTION

The structure of equations (I) will help in solving the word equations. Instead of the equation $a_{k+r} = g_r$ in (I) let us consider its associated equation $r = 1,\ldots,g-k$

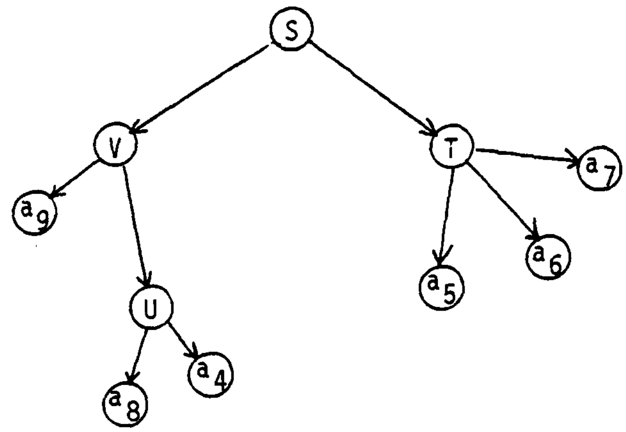
$$\phi_{(r)} = a_{i_1}\ldots a_{i_j}\ \underline{a_{k+r}}\ b_1 \ldots b_j$$

as determined in the proof of Theorem 1. Thus $a_{i_1} \ldots a_{i_j}$ denotes a descent down the spanning tree T, a_{k+r} the unknown in (I), $b_1 \ldots b_j$ a shortest path from $w(a_{k+r})$ to F.

From T we shall construct a new tree T' by adding leaves to T as follows. The new leaves will be labelled $a_{j+1},\ldots,a_q$ and will be directed respectively to the nodes

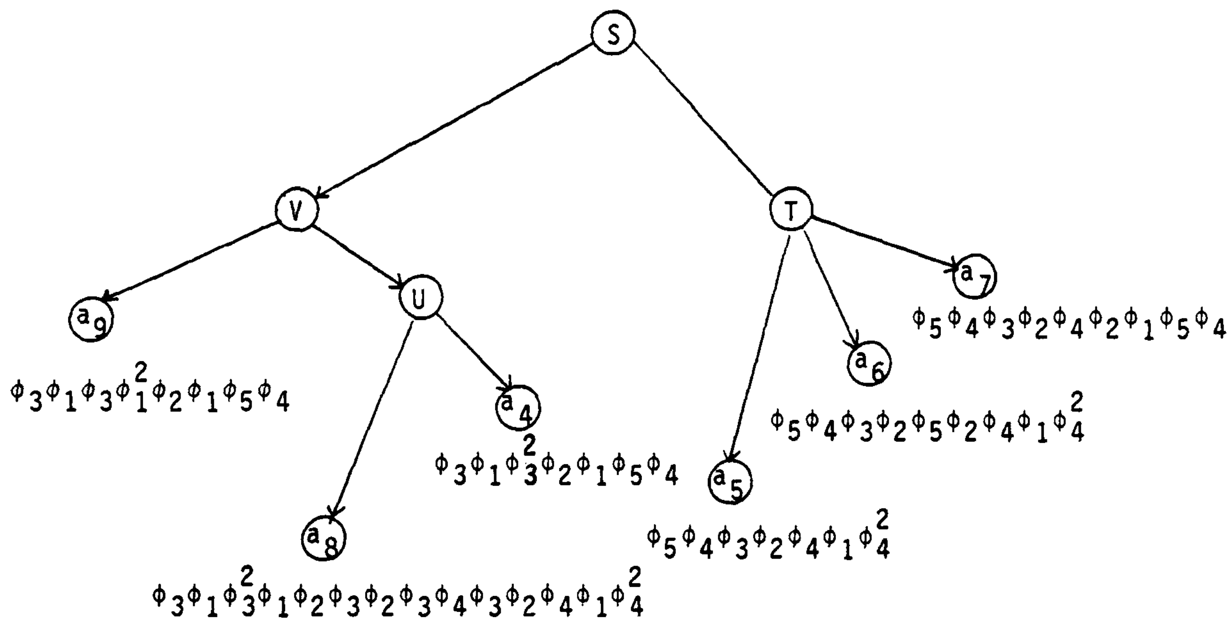
$$\alpha(a_{j+1}),\ldots,\alpha(a_q) .$$

Thus the spanning tree T of Figure 1 becomes T' in Figure 4. If we consider $TC(T',x)$ where $x \in (\Phi^*)^{q-k}$ $x = (\phi_{(1)},\ldots,\phi_{(q-k)})$ is the vector of observed sentences from Φ^*, then obviously the set of feasible points $a_1,\ldots,a_k$ are in $TC(T',x)\Big|_{a_1,\ldots,a_k}$; that is, $TC(T',x)$ restricted to the arcs $a_1,\ldots,a_k$. In some examples it turns out that a feasible point can be discovered by computing $\max[TC(T',x)]$, but this is not always the case. Consider Figures 5 and 6.

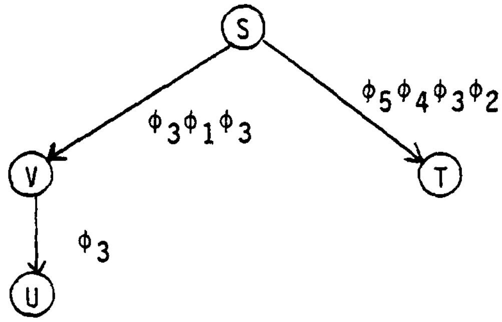


T'

<u>Figure 4</u>.



<u>Figure 5</u>.



$$\max TC(T',x)\Big|_{a_1,a_2,a_3}$$

Figure 6.

Figure 6 gives $\max TC(T',x)\Big|_{a_1,a_2,a_3}$, a feasible point (which is easily verified).

Figure 7 gives an example of a case where $\max TC(T',x)\Big|_{a_i \varepsilon T}$ is not a feasible point.

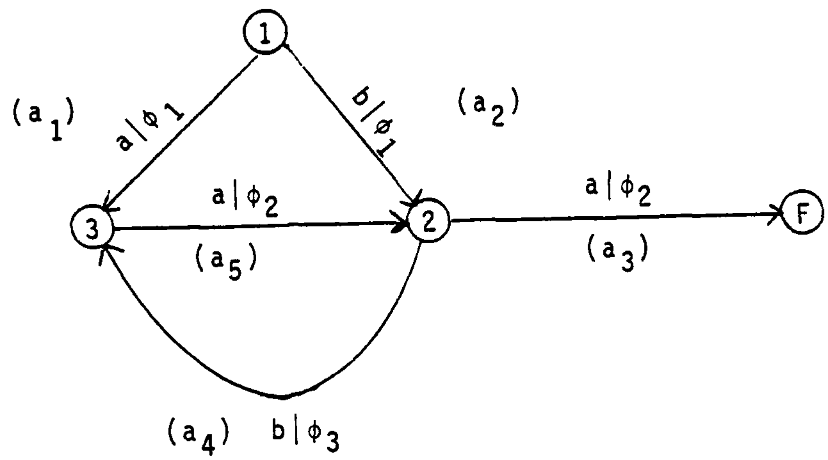
An obvious necessary condition, in addition to the feasible points being in $TC(T',x)\Big|_{a_i \varepsilon T}$, is

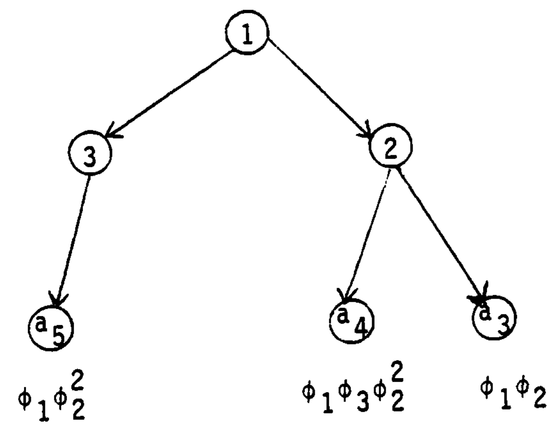
$$|\phi_{(r)}| = \left|a_{i_1}\right| + \ldots + \left|a_{i_j}\right| + |a_{\underline{k+r}}| + |b_1| + \ldots + |b_j| .$$

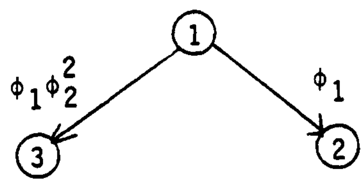
Note for the example in Figure 7, if we let $|a_i| = x_i$ then

$$\begin{aligned} x_2 + \underline{x_3} \quad\quad\quad\; &= 2 \\ x_1 + \underline{x_5} + x_3 \quad\; &= 3 \\ x_2 + \underline{x_4} + x_5 + x_3 &= 4 . \end{aligned}$$

If $x_1 = 3$ and $x_2 = 1$, as we have in the $\max TC(T',x)\Big|_{a \varepsilon T}$ solution, then there is no (x_3, x_4, x_5) *non-negative solution*.







$\max TC(T',x)\Big|_{a_i \in T}$

Figure 7.

As before, we denote the word equation for the variable a_{k+r} by $(a_{k+r} \in A^{(j)})$

$$a_{i_1} \dots a_{i_\ell} \underline{a_{k+r}} b_1 \dots b_j = \phi_{(r)} .$$

Let us now assume that $b_1 \dots b_j$ (a shortest path from $w(a_{k+r})$ to F) is chosen so that it is a suffix of a previously defined walk.

<u>THEOREM</u>: A sufficient condition for an assignment of arcs $a \in T$ to values in Φ^* to be feasible is that it satisfies

$$\max TC(T',\phi)$$

subject to

$$(*) \quad |w_{(j)}| = \phi_{(j)}$$

where $w_{(j)}$ is the walk from S to F corresponding to the variable a_{k+j} .

<u>Proof</u>: Let $\hat{\phi}(a)$, $a \in A(G)$, be the "true" unknown syntactic translation, so for $r = 1,\dots, g-k$

$$\hat{\phi}(a_{i_1}) \dots \hat{\phi}(a_{i_\ell})\hat{\phi}(a_{k+r})\hat{\phi}(b_1^{(r)}) \dots \hat{\phi}(b_j^{(r)}) .$$

Let $\phi(a)\Big|_{a \in T}$ be the assignment determined by the criteria stated in the theorem.

We claim that for each $s = 1,\dots, \ell$

$$\phi(a_{i_s}) \dots \phi(a_{i_\ell}) \; \phi(a_{k+r}) \dots \phi(b_j)$$

is a suffix of

$$\hat{\phi}(a_{i_s}) \dots \hat{\phi}(a_{i_\ell}) \; \hat{\phi}(a_{k+r}) \dots \hat{\phi}(b_j) .$$

If this were not true, then we would have, for some s ,

$$\phi(a_{i_1}) \dots \phi(a_{i_{s-1}})$$

being a proper prefix of

$$\hat{\phi}(a_{i_1}) \ldots \hat{\phi}(a_{i_{s-1}})$$

and this contradicts maximality.

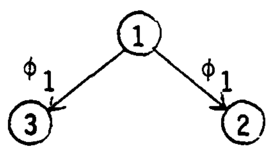
Consequently, $\phi(b_1) \ldots \phi(b_j)$ is a suffix of $\phi_{(r)}$ (by induction, $b_1 \ldots b_j$ is of the form $a'_{i_s} \ldots a'_{i_\ell}\ a'_{k+r}\ b'_1 \ldots b'_j$ for a previously computed walk) $\phi(a_{i_1}) \ldots \phi(a_{i_\ell})$ is a prefix of $\phi_{(r)}$, so by (*) we have a solution in Φ^* of $\phi(a_{k+r})$. []

The example of Figure 7 shows that

$$\begin{cases} x_2 + x_3 = 2 \\ x_1 + x_5 + x_3 = 3 \\ x_2 + x_4 + x_5 + x_3 = 4 \end{cases}$$

$$\Rightarrow (x_1,x_2) \in \{(0,0),(0,1),(1,0),(1,1),(1,2)\} .$$

$(x_1,x_2) = (1,1)$ corresponds to



$$\max TC(T',x)\Big|_{a \in T}$$

subject to (*)

which is indeed feasible.

It is evident that we may replace $TC(T',\phi)$ with a set of inequalities, i.e., for the example in Figure 7 we must have

$$x_1 \le 3$$
$$x_2 \le 1 \ ,$$

for the example in Figure 5

$$x_1 \le 3$$
$$x_2 \le 4$$
$$x_1 + x_3 \le 5 \ .$$

REFERENCES

[1] Harrison, M. (1978). *Introduction to Formal Language Theory.* Addison-Wesley.

[2] Hopcroft, J. and Ullman, J. (1969). *Formal Languages and Their Relation to Automata.* Addison-Wesley.

[3] Harary, F. (1969). *Graph Theory.* Addison-Wesley.

[4] Thrift, P. (1979). "An Implicit Function Theorem for Group Equations Generated by a Finite Automaton," Tech. Rep. #150, Series 2, Department of Statistics, Princeton University.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER T.R.#160-Series 2	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) AN INDUCTION THEOREM FOR DISCOVERING SYNTACTIC TRANSLATIONS		5. TYPE OF REPORT & PERIOD COVERED Technical Report 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Philip R. Thrift		8. CONTRACT OR GRANT NUMBER(s) N00014-79-C-0322
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Statistics Princeton University Princeton, N. J. 08540		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research (Code 436) Arlington, Virginia 22217		12. REPORT DATE January 1980 13. NUMBER OF PAGES 17
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Given an input-output sequence of syntactic translations of sentences generated by a deterministic finite state grammar G into Σ^*, a method is given for discovering the function which maps productions of G into Σ^* that gives rise to the observed translation.